

Instructions for launching OpenNebula instances within fermicloud

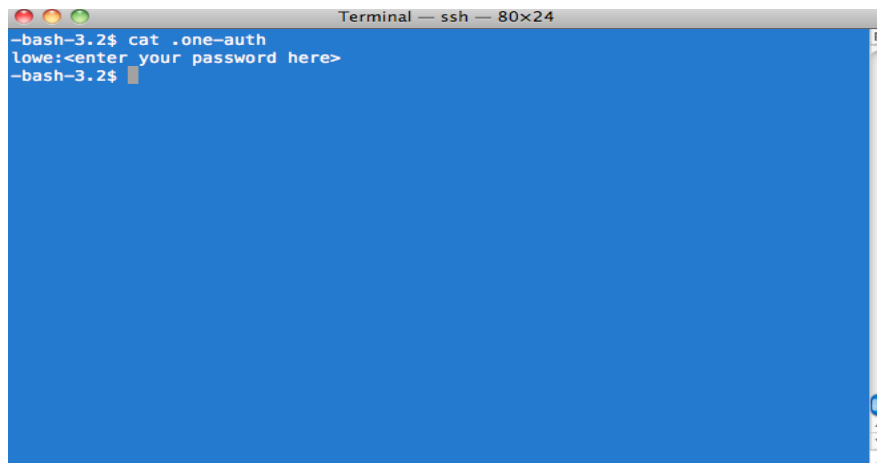
These instructions are designed to instruct users how to launch cloud VMs within OpenNebula.

Prior to logging into fcl002 please contact the fermicloud team and obtain an account for fermicloud. This account will be used to log into the cloud, manage your templates for cloud instances, start and stop your VM instances.

User Environment.

Before launching OpenNebula instances we have to setup the user environment.

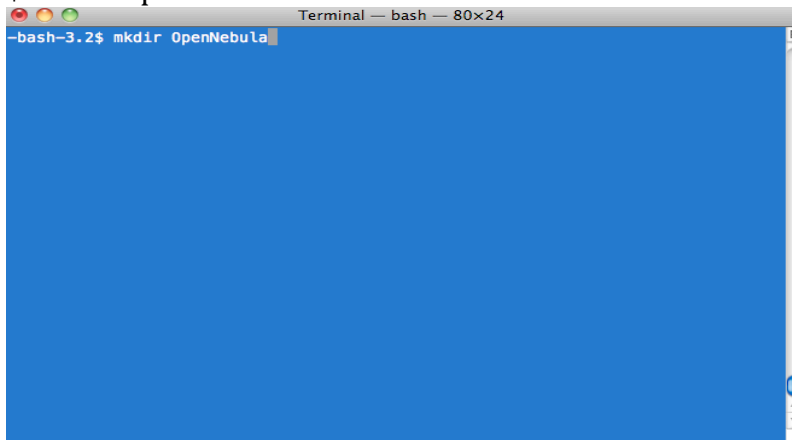
1. ssh into fcl002 as yourself

A terminal window titled "Terminal — ssh — 80x24" with a blue background. The prompt is "-bash-3.2\$". The user has entered "cat .one-auth", and the output is "lowe:<enter your password here>". The prompt is now "-bash-3.2\$".

```
Terminal — ssh — 80x24
-bash-3.2$ cat .one-auth
lowe:<enter your password here>
-bash-3.2$
```

2. Create a directory called OpenNebula beneath your home directory

\$ mkdir OpenNebula

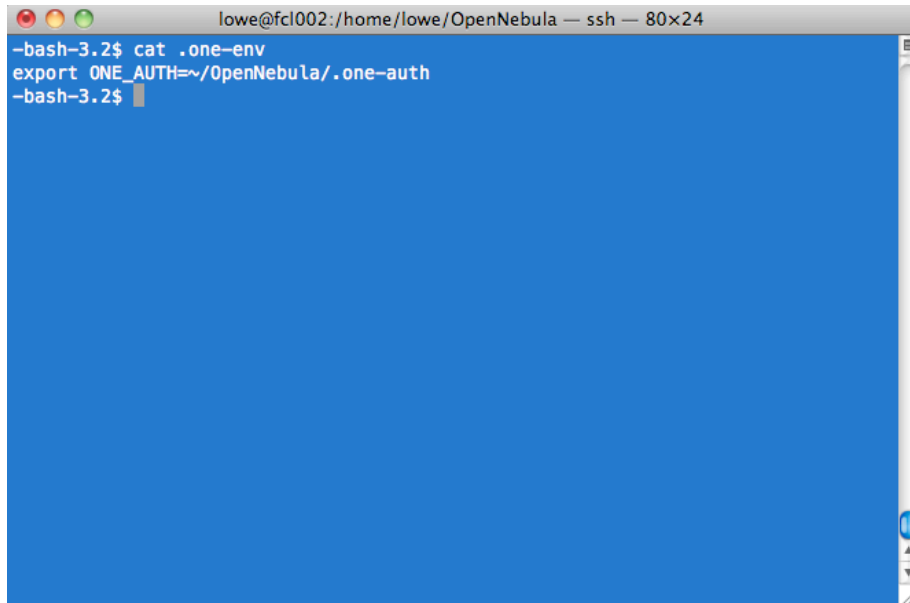
A terminal window titled "Terminal — bash — 80x24" with a blue background. The prompt is "-bash-3.2\$". The user has entered "mkdir OpenNebula".

```
Terminal — bash — 80x24
-bash-3.2$ mkdir OpenNebula
```

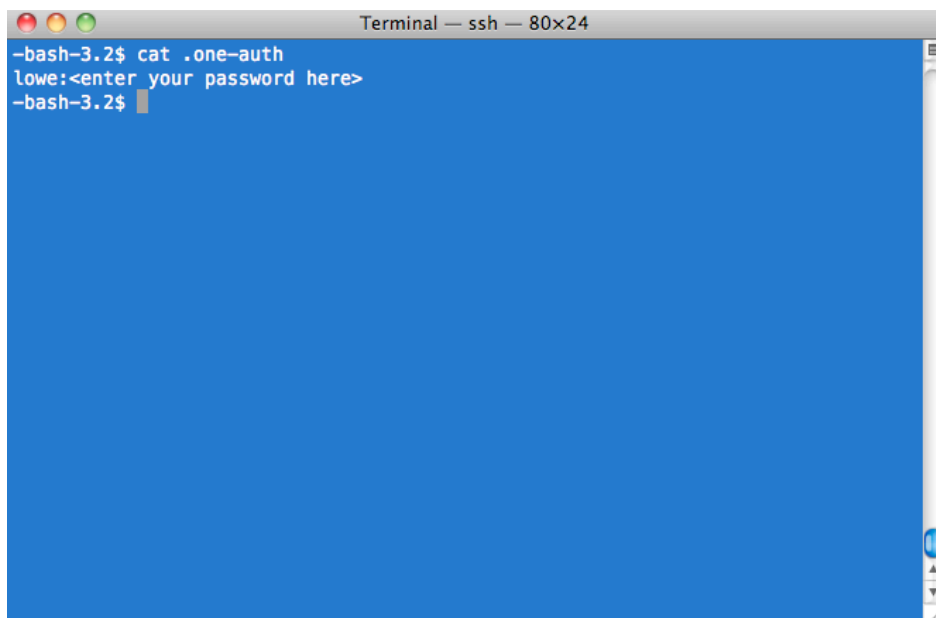
3. Change directory to the ~/OpenNebula directory and create the following files:

- .one-env
- .one-auth

.one-env should look as follows:

A terminal window titled 'lowe@fcl002:/home/lowe/OpenNebula — ssh — 80x24'. The terminal shows the command 'cat .one-env' being executed, which outputs 'export ONE_AUTH=~/.one-auth'. The prompt is '-bash-3.2\$'.

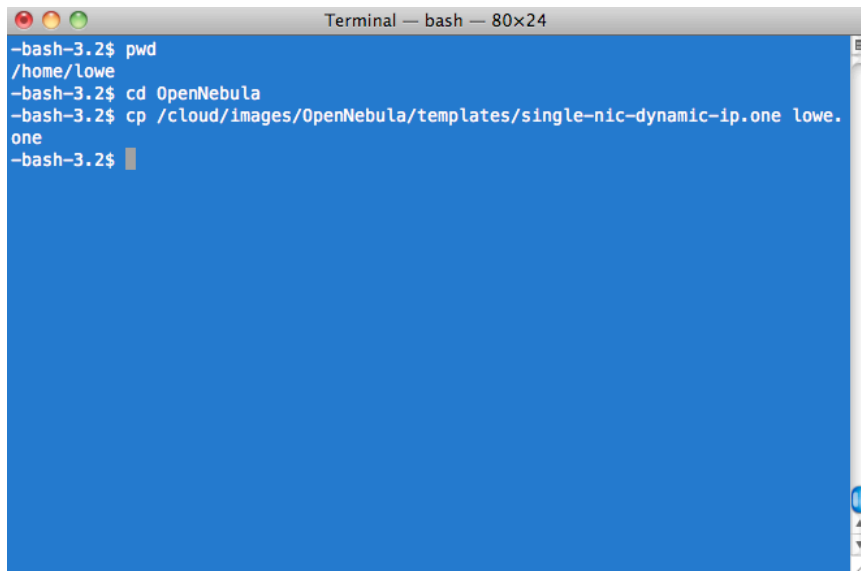
.one-auth should appear as follows: (NOTE: Be sure to use your password)

A terminal window titled 'Terminal — ssh — 80x24'. The terminal shows the command 'cat .one-auth' being executed, which prompts 'lowe:<enter your password here>'. The prompt is '-bash-3.2\$'.

NOTE: This password is the password for your OpenNebula account, if you do not have an OpenNebula account please contact someone on the Fermicloud team to obtain an account.

Template Management and customizations.

1. If you haven't already done so, change directories to `$HOME/OpenNebula`. Then, copy one of the sample templates from `/cloud/images/OpenNebula/templates`. In our example we will copy the template for a dynamically assigned public IP address.

A terminal window titled "Terminal — bash — 80x24" with a blue background. It shows a series of commands and their outputs:

```
-bash-3.2$ pwd
/home/lowe
-bash-3.2$ cd OpenNebula
-bash-3.2$ cp /cloud/images/OpenNebula/templates/single-nic-dynamic-ip.one lowe.
one
-bash-3.2$
```

The template examples cover three types of VMs

1. A dynamically assigned public IP address (from fermicloud's pool of public IP addresses). See template `/cloud/images/OpenNebula/templates/single-nic-dynamic-ip.one`
2. A dynamically assigned public IP address and a dynamically assigned private address (again from fermicloud's pools of public and private IP addresses). See template `/cloud/images/OpenNebula/templates/dual-nic-dynamic-ips.one`
3. A statically assigned IP address assigned by fermicloud staff. See template `/cloud/images/OpenNebula/templates/static-ip.one`

****If you have a scenario where you need to have a static IP for your VM,**

please contact someone on the Fermicloud team to have your template modified with the appropriate IP address

After copying the template file to your OpenNebula directory, create a k5login file in your OpenNebula directory. Your k5login file should contain kerberos principals users allowed root access to your VM.

Below is an example of a supported k5login file

```
- cloudadmin/cron/fermigrid.fnal.gov@FNAL.GOV
timm@FNAL.GOV
chadwick@FNAL.GOV
yocum@FNAL.GOV
neha@FNAL.GOV
lowe@FNAL.GOV
<Additional Kerberos principals are added below>
```

NOTE: To received support from Fermicloud staff with your VM, please ensure that you include all of the Kerberos principals above within your .k5login. Failing to do so will not allow fermicloud staff to access your VM as root, thus preventing them from supporting your instance.

ADDITIONAL NOTE: While typically the k5login file is named .k5login, a hidden file, do not name the file as it will appear on the system. Our script will take care of that. Naming the file .k5login will break the init script.

Once you have your template, change the following entry within your template.

```
NAME = <username>
files = "/cloud/images/OpenNebula/templates/init.sh /home/<username>/OpenNebula/k5login",
```

Change the <username> to your username the lines should therefore look like the following:

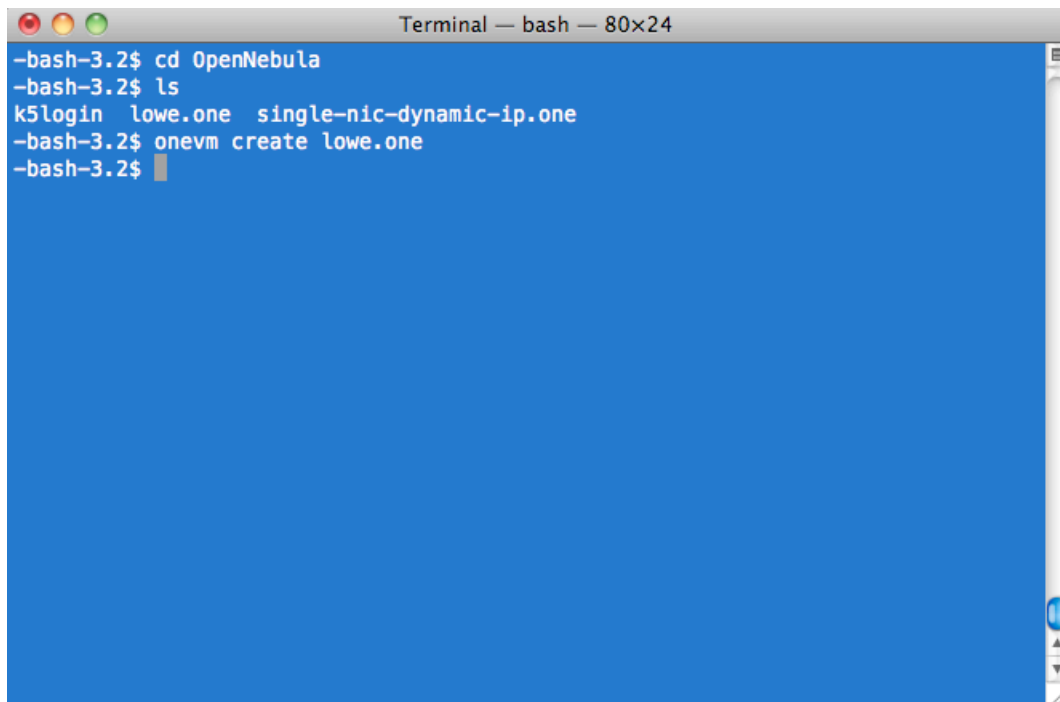
```
NAME = lowe
files = "/cloud/images/OpenNebula/templates/init.sh /home/lowe/OpenNebula/k5login",
```

Once you have modified your template and your k5login file you are ready to launch your image.

Launching your Image

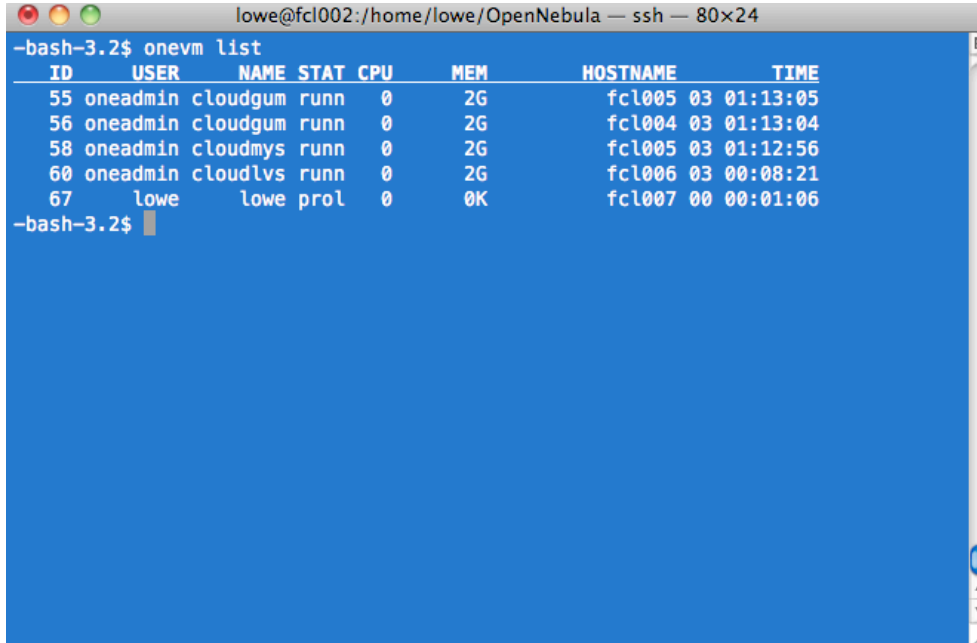
The following procedure will guide you through the process of launching your VM instance. While there are numerous options that one may add to the `onevm` command and a couple of other methods of deploying an instance, for now we will stick to these guidelines and as time progresses we will introduce other methods of deploying images within the environment.

1. If you have not already done so, login to `fcl002` and change directory to `$HOME/OpenNebula`, as previously illustrated.
2. Run `onevm create <template name>` as illustrated below

A terminal window titled "Terminal — bash — 80x24" with a blue background. The window shows a series of commands and their outputs. The first command is `cd OpenNebula`, followed by `ls` which lists `k5login`, `lowe.one`, and `single-nic-dynamic-ip.one`. The third command is `onevm create lowe.one`, and the prompt returns to `-bash-3.2$`.

```
-bash-3.2$ cd OpenNebula
-bash-3.2$ ls
k5login  lowe.one  single-nic-dynamic-ip.one
-bash-3.2$ onevm create lowe.one
-bash-3.2$
```

3. To check the status of your vm execute the following command, `onevm list`



```
-bash-3.2$ onevm list
  ID   USER   NAME  STAT  CPU   MEM   HOSTNAME   TIME
  ---  ---
  55  oneadmin cloudgum runn   0    2G   fcl005 03 01:13:05
  56  oneadmin cloudgum runn   0    2G   fcl004 03 01:13:04
  58  oneadmin cloudmys runn   0    2G   fcl005 03 01:12:56
  60  oneadmin cloudlvs runn   0    2G   fcl006 03 00:08:21
  67   lowe    lowe prol    0    0K   fcl007 00 00:01:06
-bash-3.2$
```

Beneath the status column you will see a couple of different states here are the meanings of the various states of your instance that you may see:

VM States:

- * pend pending
- * hold VM on hold (not runnable)
- * stop stopped
- * susp suspended
- * done finished
- * prol prolog
- * boot booting
- * runn running
- * migr migrating
- * save saving the VM to disk
- * epil epilog
- * shut shutting down
- * fail failed

Prolog is the state that your instances is in while copying the image to the respective server that will host the VM instances.

If your image fails to launch, look in the `/var/log/one` directory. For example, if you have vm id #67, the log of the failure will be `67.log`.

For the case of the variable-IP VM, you can determine what IP it is going to have by doing the following

```
onevm show 67
```

```
IP=131.225.154.149
```

Then you can ssh into your machine as root or as yourself.

To stop the VM do `onevm stop`

```
$ onevm stop <VM_ID>
```